# Sentiment Analysis

Avram Ștefan Alexandru, Anghel Ana, Manolachi Tiberiu-Andrei, Voiculescu Ioana-Daria

15 May 2024

# Table of contents

# Introduction

- What is sentiment analysis?
- Why is it important?
- Project objectives

# Introduction

- What is sentiment analysis?
- Why is it important?
- Project objectives

# Introduction

- What is sentiment analysis?
- Why is it important?
- Project objectives

# Data Acquisition

- Data source: Twitter API
- About our data
- Data preprocessing

# Data Acquisition

- Data source: Twitter API
- About our data
- Data preprocessing

# Data Acquisition

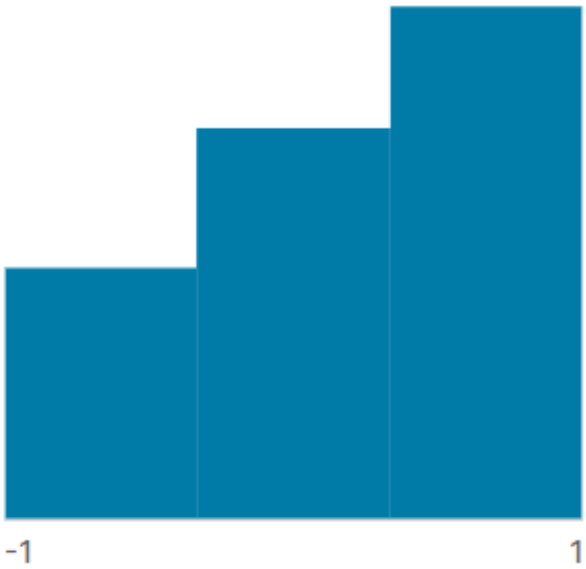
- Data source: Twitter API
- About our data
- Data preprocessing

# Data Acquisition

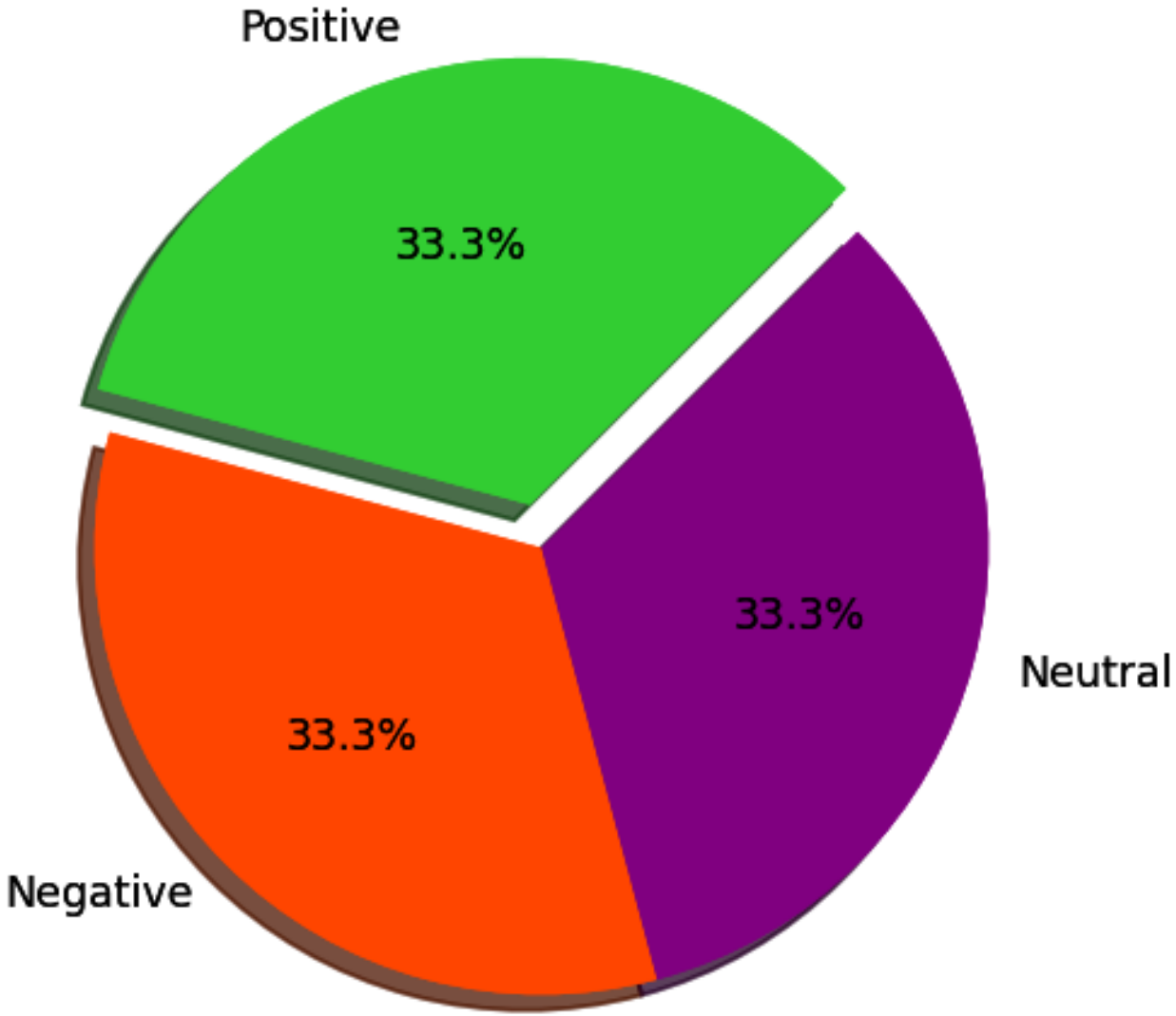
- Data source: Twitter API
- About our data
- Data preprocessing

**# category**

Describes the Actual Sentiment of the Respective Tweet Ranging from -1 to 1




| | | |
|---|---|---|
| Valid | 163k | 100% |
| Mismatched | 0 | 0% |
| Missing | 10 | 0% |
| Mean | 0.23 | |
| Std. Deviation | 0.78 | |
| Quantiles | -1 | Min |
| | 0 | 25% |
| | 0 | 50% |
| | 1 | 75% |
| | 1 | Max |


Positive
33.3%
33.3%
Neutral
33.3%
Negative

# Model Selection

## Approaches

- Linear Regression
- SVM
- KNN
- Naive-Bayes
- Decison Tree

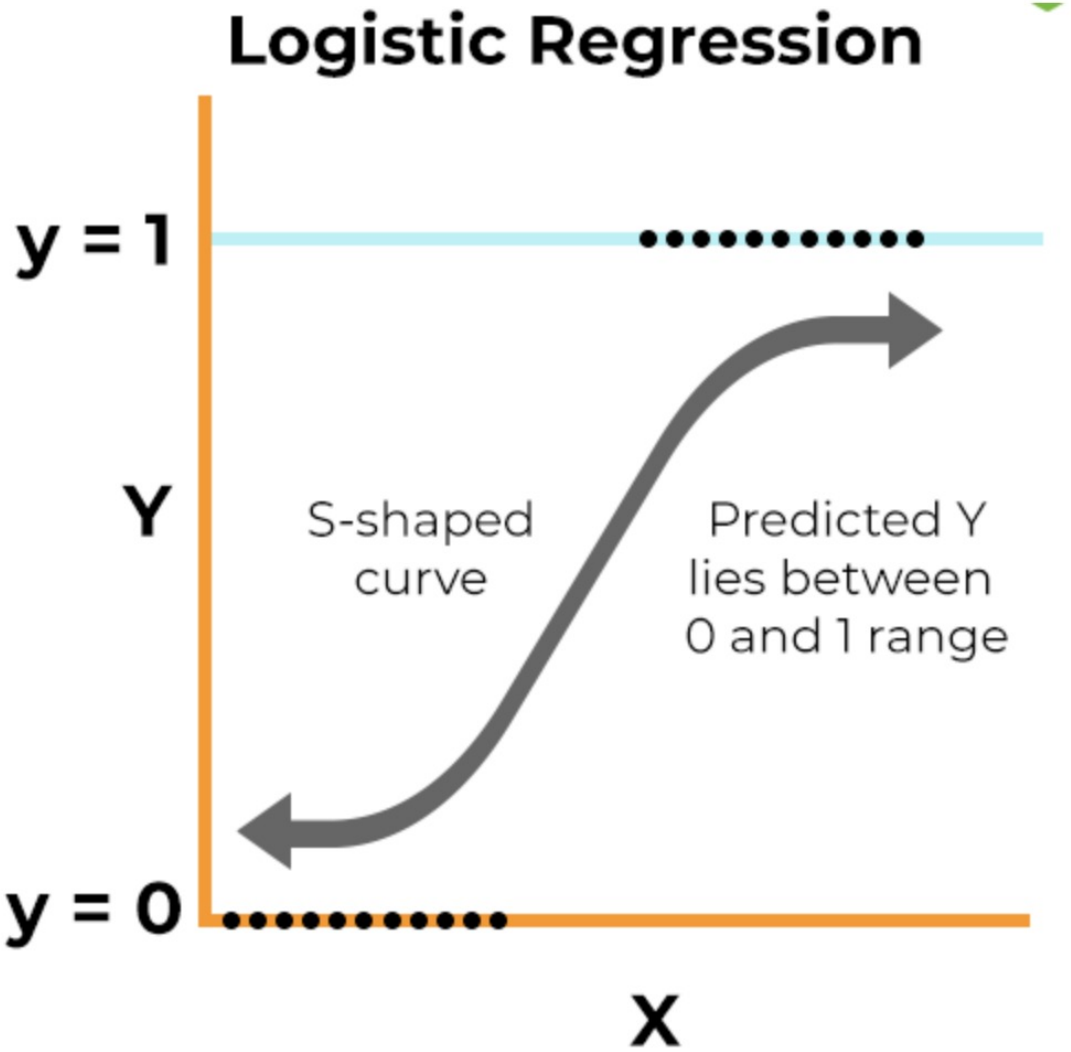
- Popular machine learning algorithm widely used for classification tasks
- How: learns a linear relationship between **features** (TF-IDF vectors from text data) and a **target variable** (the sentiment)
- Training: the model estimates coefficients for each feature that influence the probability of a data point belonging to a specific class
- Sentiment Analysis: the model predicts the most likely sentiment (positive, negative, neutral) for a new unseen text sample based on the learned coefficients and the features extracted from the text

- Popular machine learning algorithm widely used for classification tasks
- How: learns a linear relationship between **features** (TF-IDF vectors from text data) and a **target variable** (the sentiment)
- Training: the model estimates coefficients for each feature that influence the probability of a data point belonging to a specific class
- Sentiment Analysis: the model predicts the most likely sentiment (positive, negative, neutral) for a new unseen text sample based on the learned coefficients and the features extracted from the text

- Popular machine learning algorithm widely used for classification tasks
- How: learns a linear relationship between **features** (TF-IDF vectors from text data) and a **target variable** (the sentiment)
- Training: the model estimates coefficients for each feature that influence the probability of a data point belonging to a specific class
- Sentiment Analysis: the model predicts the most likely sentiment (positive, negative, neutral) for a new unseen text sample based on the learned coefficients and the features extracted from the text

- Popular machine learning algorithm widely used for classification tasks
- How: learns a linear relationship between **features** (TF-IDF vectors from text data) and a **target variable** (the sentiment)
- Training: the model estimates coefficients for each feature that influence the probability of a data point belonging to a specific class
- Sentiment Analysis: the model predicts the most likely sentiment (positive, negative, neutral) for a new unseen text sample based on the learned coefficients and the features extracted from the text

## Logistic Regression



# Advantages

- **Interpretability**: we can understand which features contribute most to predicting positive, neutral or negative sentiment
- **Simplicity**: it is a relatively simple algorithm
- **Efficiency**: computationally efficient to train and can handle large datasets effectively

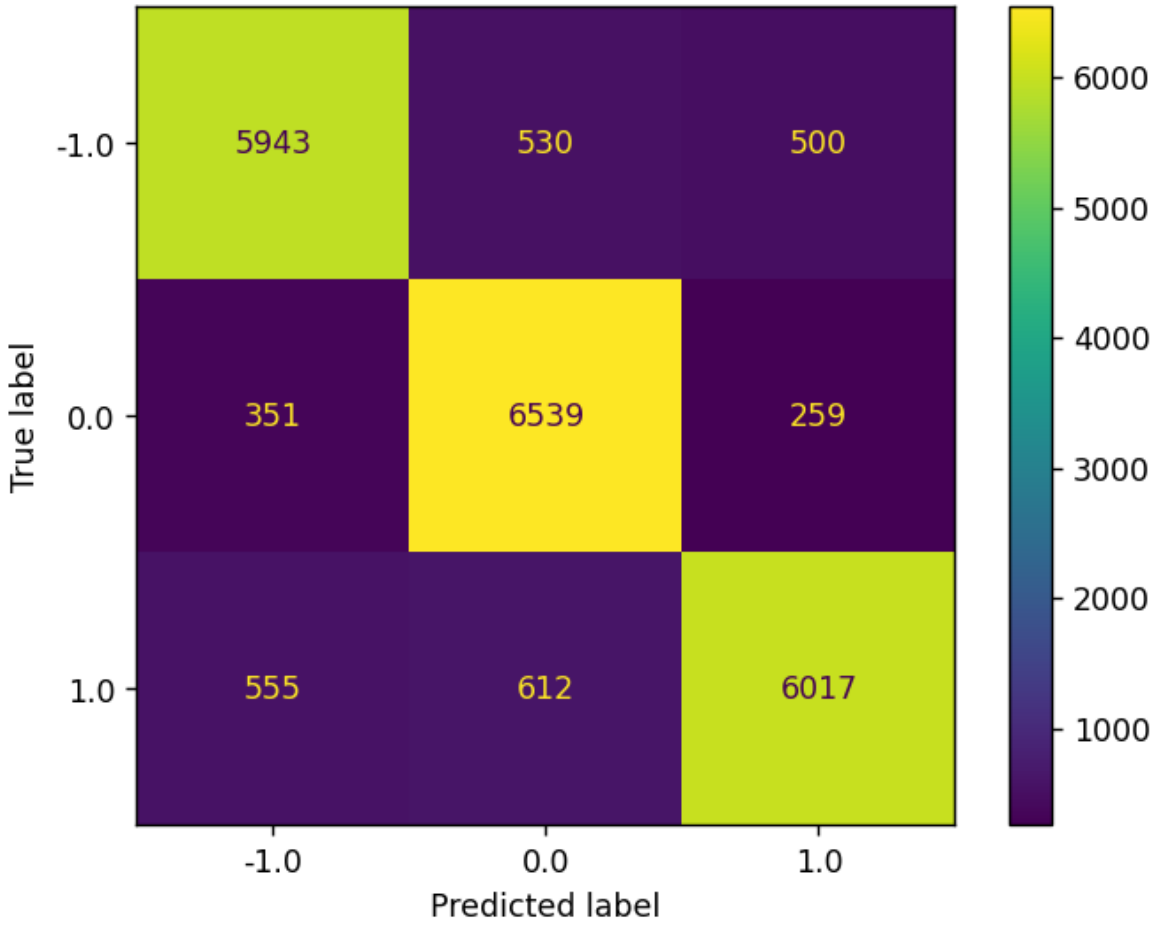
# Advantages

- **Interpretability**: we can understand which features contribute most to predicting positive, neutral or negative sentiment
- **Simplicity**: it is a relatively simple algorithm
- **Efficiency**: computationally efficient to train and can handle large datasets effectively

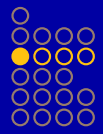
# Advantages

- **Interpretability**: we can understand which features contribute most to predicting positive, neutral or negative sentiment
- **Simplicity**: it is a relatively simple algorithm
- **Efficiency**: computationally efficient to train and can handle large datasets effectively

# Performance



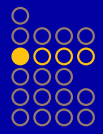
- While Logistic Regression learns a linear decision boundary, SVMs aim to find a hyperplane in the feature space that **maximizes** the margin between the data points (the support vectors) belonging to different classes
- The margin = **confidence** of classification
- In Sentiment Analysis: SVMs learn a hyperplane that effectively separates positive, neutral and negative sentiment data points based on extracted features (TF-IDF vectors)
- New text data is then classified based on which side of the hyperplane it falls on



## SVM

- While Logistic Regression learns a linear decision boundary, SVMs aim to find a hyperplane in the feature space that **maximizes** the margin between the data points (the support vectors) belonging to different classes
- The margin = **confidence** of classification
- In Sentiment Analysis: SVMs learn a hyperplane that effectively separates positive, neutral and negative sentiment data points based on extracted features (TF-IDF vectors)
- New text data is then classified based on which side of the hyperplane it falls on

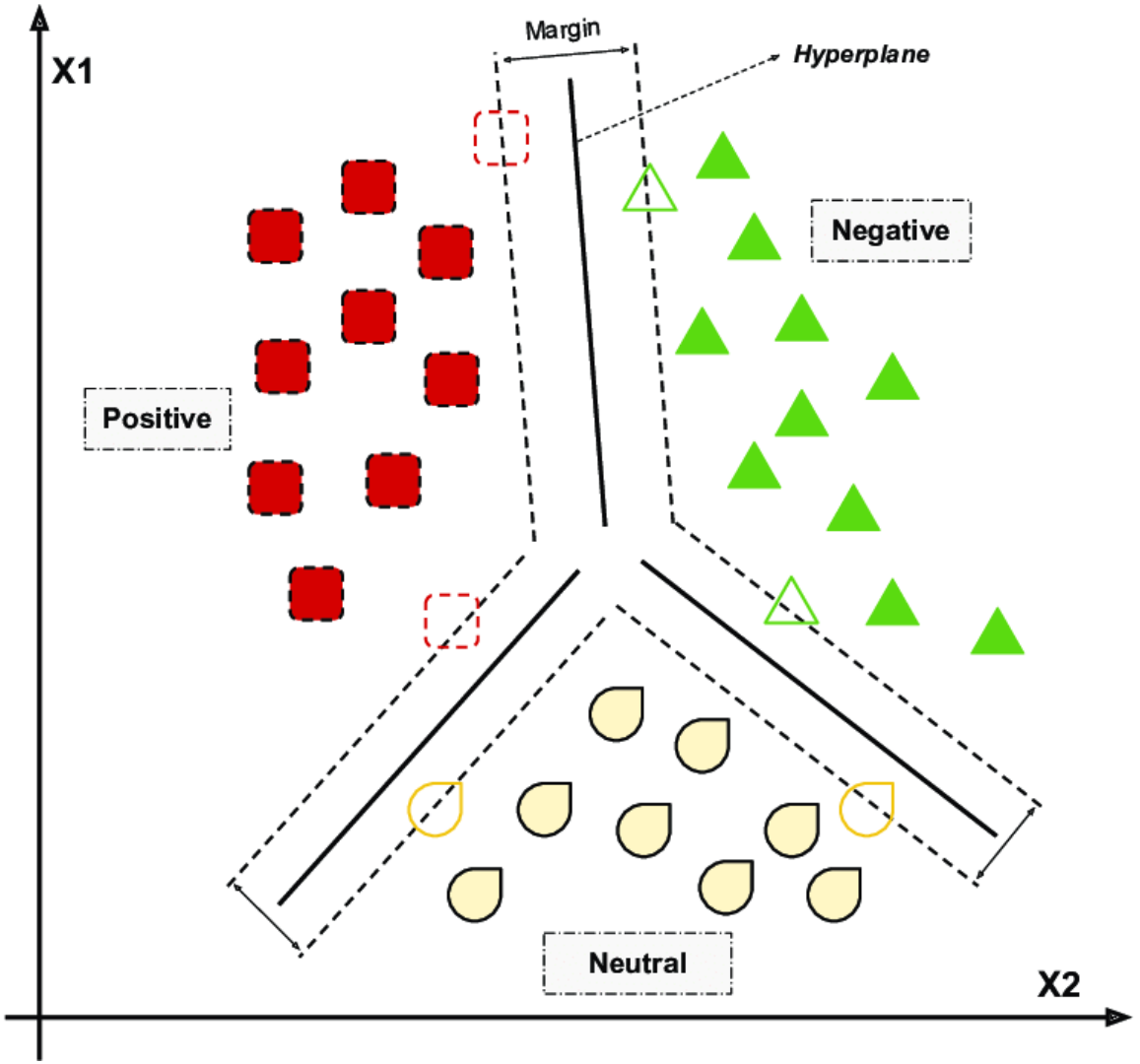
- While Logistic Regression learns a linear decision boundary, SVMs aim to find a hyperplane in the feature space that **maximizes** the margin between the data points (the support vectors) belonging to different classes
- The margin = **confidence** of classification
- In Sentiment Analysis: SVMs learn a hyperplane that effectively separates positive, neutral and negative sentiment data points based on extracted features (TF-IDF vectors)
- New text data is then classified based on which side of the hyperplane it falls on



## SVM

- While Logistic Regression learns a linear decision boundary, SVMs aim to find a hyperplane in the feature space that **maximizes** the margin between the data points (the support vectors) belonging to different classes
- The margin = **confidence** of classification
- In Sentiment Analysis: SVMs learn a hyperplane that effectively separates positive, neutral and negative sentiment data points based on extracted features (TF-IDF vectors)
- New text data is then classified based on which side of the hyperplane it falls on

## SVM



# Advantages

- **High Accuracy**: SVMs are known for achieving
- **Effective with high-dimension data**: SVMs perform well, even in high-dimensional feature spaces (commonly encountered in NLP)
- **Robust to noise**: SVMs are relatively insensitive to irrelevant data points (that might impact other algorithms)

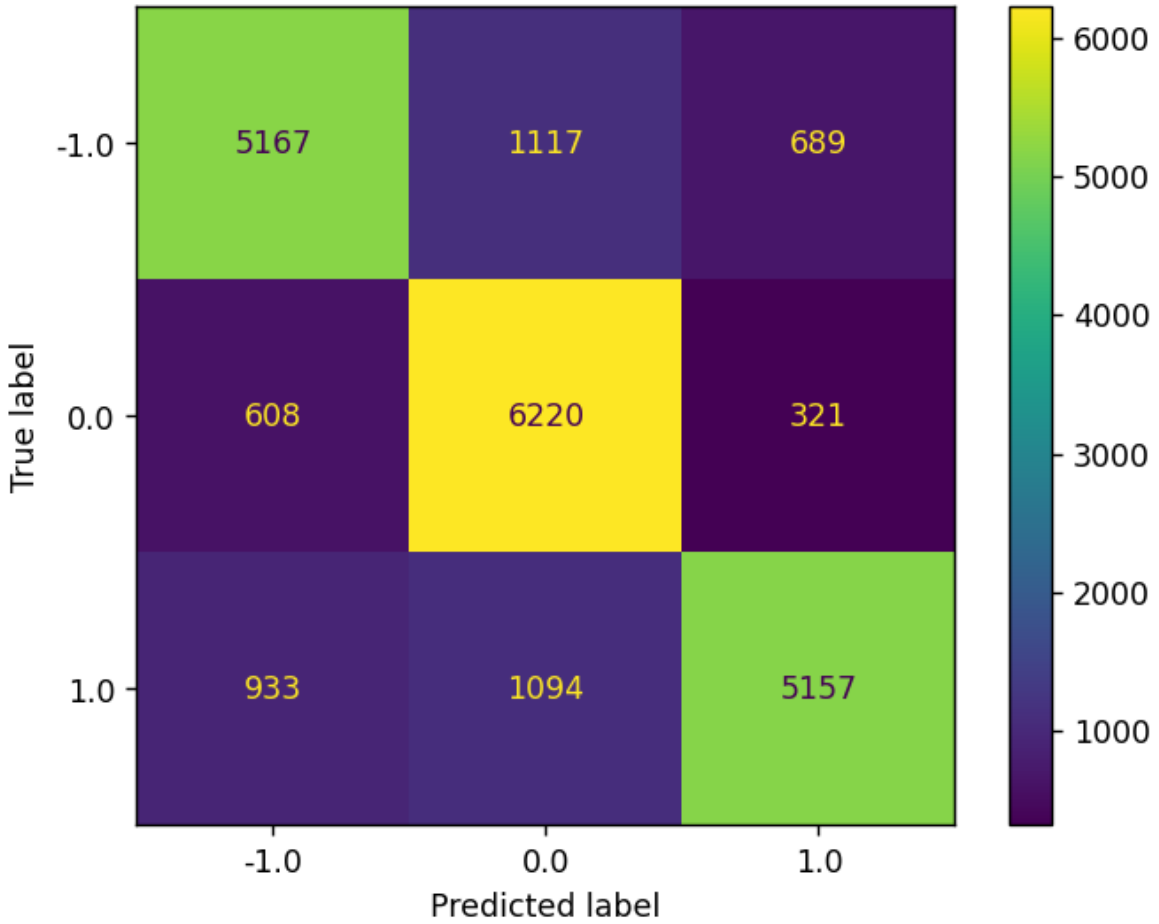
# Advantages

- **High Accuracy**: SVMs are known for achieving
- **Effective with high-dimension data**: SVMs perform well, even in high-dimensional feature spaces (commonly encountered in NLP)
- **Robust to noise**: SVMs are relatively insensitive to irrelevant data points (that might impact other algorithms)

# Advantages

- **High Accuracy**: SVMs are known for achieving
- **Effective with high-dimension data**: SVMs perform well, even in high-dimensional feature spaces (commonly encountered in NLP)
- **Robust to noise**: SVMs are relatively insensitive to irrelevant data points (that might impact other algorithms)

# Performance



## KNN

- Classifies based on nearest neighbors in training data (similar features).
- Training: Stores the entire training dataset. It doesn't learn a model by fitting coefficients, but rather memorizes the data points and their corresponding sentiment labels. This essentially creates a reference set for comparison during prediction.
- Sentiment Analysis: New text assigned sentiment based on majority vote of its k nearest neighbors.

- Classifies based on nearest neighbors in training data (similar features).
- Training: Stores the entire training dataset. It doesn't learn a model by fitting coefficients, but rather memorizes the data points and their corresponding sentiment labels. This essentially creates a reference set for comparison during prediction.
- Sentiment Analysis: New text assigned sentiment based on majority vote of its k nearest neighbors.

- Classifies based on nearest neighbors in training data (similar features).
- Training: Stores the entire training dataset. It doesn't learn a model by fitting coefficients, but rather memorizes the data points and their corresponding sentiment labels. This essentially creates a reference set for comparison during prediction.
- Sentiment Analysis: New text assigned sentiment based on majority vote of its k nearest neighbors.

# Advantages

- **Simple and intuitive**: KNN is a very easy algorithm to understand and implement. The core idea of finding similar neighbors is easy and doesn't involve complex mathematical models.
- **Effective with high-dimensional data**: KNN can handle high-dimensional data without significant performance degradation.
- **No assumptions about data distribution**: Unlike some algorithms that require specific assumptions about the underlying data distribution, KNN makes no such assumptions.

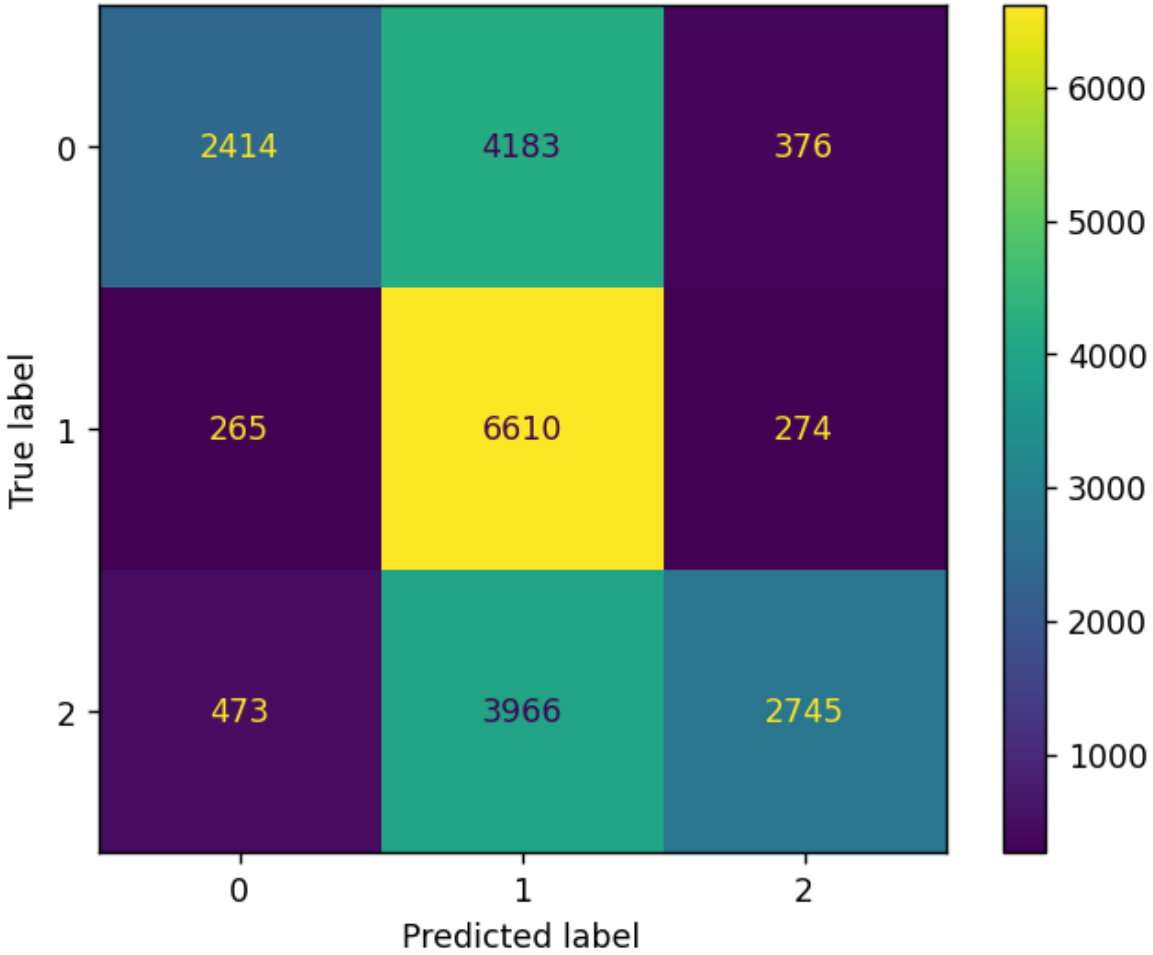
# Advantages

- **Simple and intuitive**: KNN is a very easy algorithm to understand and implement. The core idea of finding similar neighbors is easy and doesn't involve complex mathematical models.
- **Effective with high-dimensional data**: KNN can handle high-dimensional data without significant performance degradation.
- **No assumptions about data distribution**: Unlike some algorithms that require specific assumptions about the underlying data distribution, KNN makes no such assumptions.

# Advantages

- **Simple and intuitive**: KNN is a very easy algorithm to understand and implement. The core idea of finding similar neighbors is easy and doesn't involve complex mathematical models.
- **Effective with high-dimensional data**: KNN can handle high-dimensional data without significant performance degradation.
- **No assumptions about data distribution**: Unlike some algorithms that require specific assumptions about the underlying data distribution, KNN makes no such assumptions.

# Performance

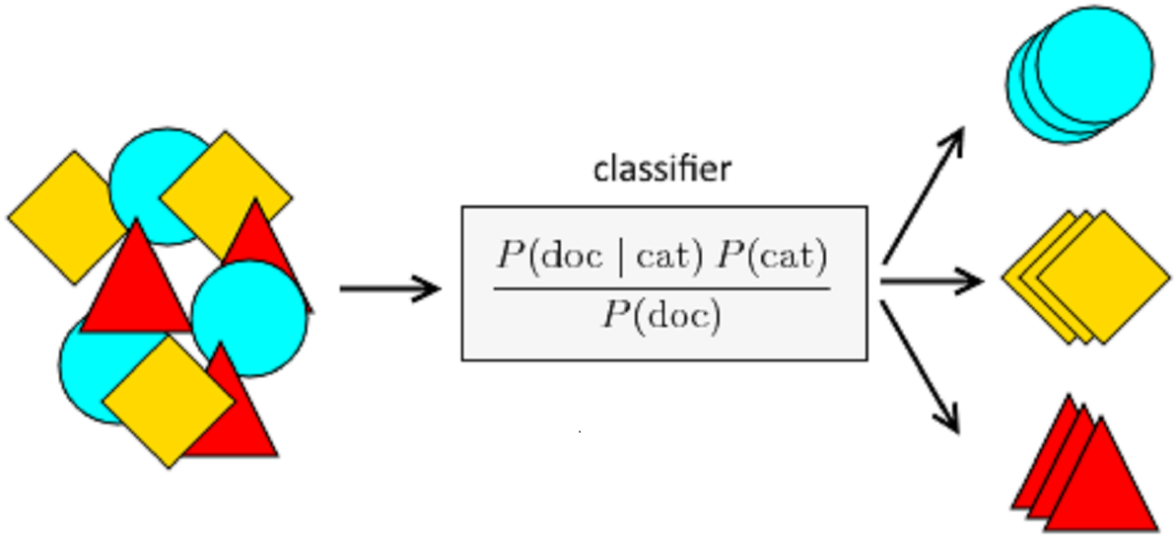


- Popular probabilistic classifier
- It assumes independence between features when predicting the class of a new data point
- The class with the highest probability is assigned to the document

- Popular probabilistic classifier
- It assumes independence between features when predicting the class of a new data point
- The class with the highest probability is assigned to the document

- Popular probabilistic classifier
- It assumes independence between features when predicting the class of a new data point
- The class with the highest probability is assigned to the document

## Naive-Bayes



# Advantages

- **Simplicity and efficiency**: relatively simple to understand and implement; computationally efficient for training on large datasets

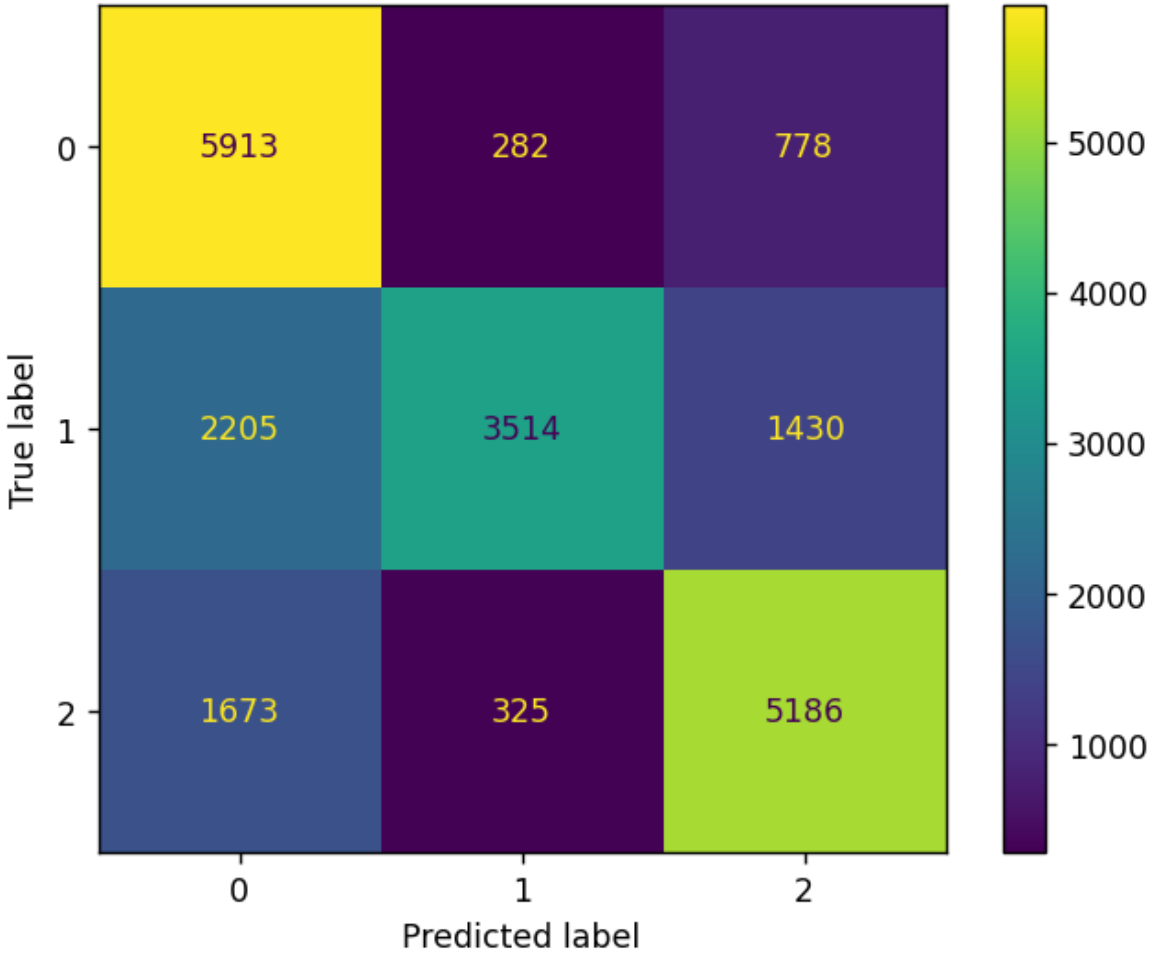
# Advantages

- **Simplicity and efficiency**: relatively simple to understand and implement; computationally efficient for training on large datasets
- **High performance**: it can achieve competitive accuracy, especially for well-structured tasks

# Advantages

- **Simplicity and efficiency**: relatively simple to understand and implement; computationally efficient for training on large datasets
- **High performance**: it can achieve competitive accuracy, especially for well-structured tasks
- **Handling high-dimensional data**: can effectively handle high-dimensional feature spaces common in NLP tasks (due to its focus on individual feature probabilities)

# Performance



- Fundamental machine learning algorithm
- How: build a tree-like structure where internal **nodes** represent features and **branches** represent decision rules based on those features
- During training, the model iteratively splits the data based on the feature that best separates the data points belonging to different classes
- The process continues until a stopping criterion is met $\rightarrow$ tree structure where leaf nodes represent the predicted sentiment class
- In Sentiment Analysis: the model analyzes the text and traverses the decision tree based on word presence or absence, ultimately reaching a leaf node that represents the predicted sentiment (positive, negative, or neutral)

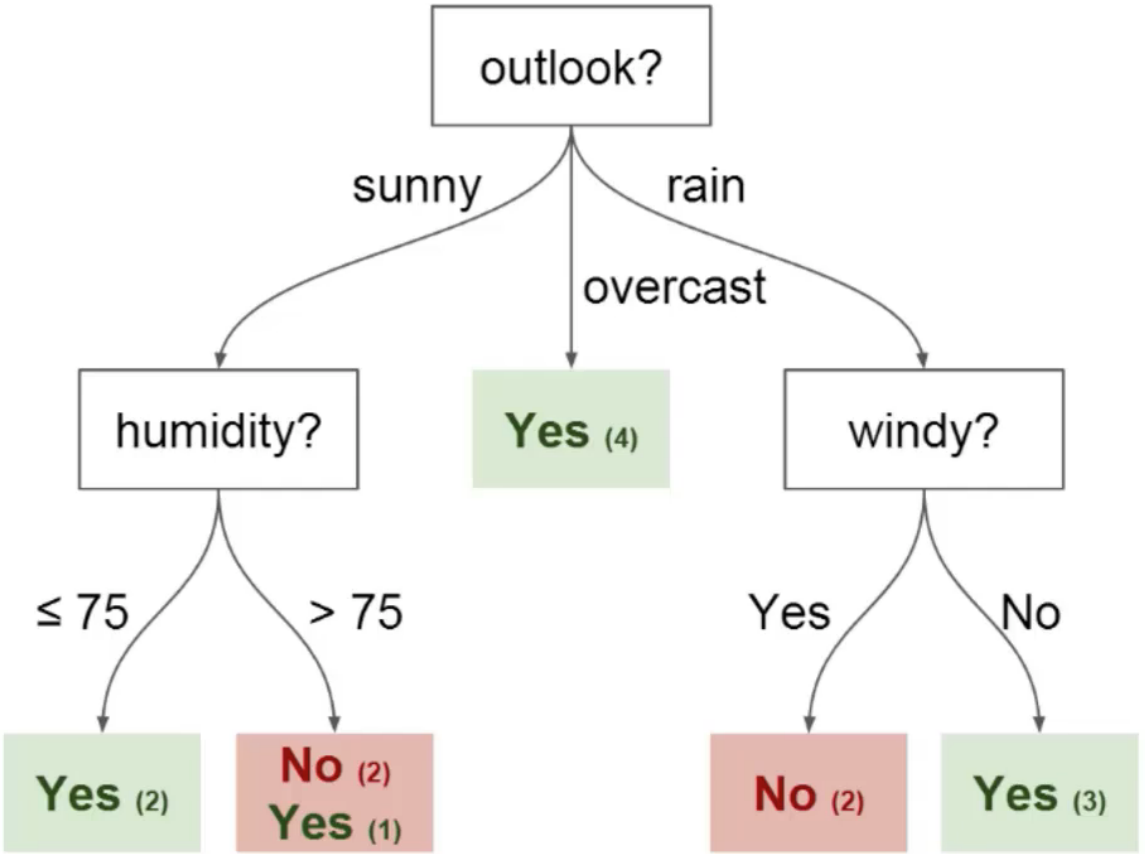
- Fundamental machine learning algorithm
- How: build a tree-like structure where internal **nodes** represent features and **branches** represent decision rules based on those features
- During training, the model iteratively splits the data based on the feature that best separates the data points belonging to different classes
- The process continues until a stopping criterion is met $\rightarrow$ tree structure where leaf nodes represent the predicted sentiment class
- In Sentiment Analysis: the model analyzes the text and traverses the decision tree based on word presence or absence, ultimately reaching a leaf node that represents the predicted sentiment (positive, negative, or neutral)

- Fundamental machine learning algorithm
- How: build a tree-like structure where internal **nodes** represent features and **branches** represent decision rules based on those features
- During training, the model iteratively splits the data based on the feature that best separates the data points belonging to different classes
- The process continues until a stopping criterion is met $\rightarrow$ tree structure where leaf nodes represent the predicted sentiment class
- In Sentiment Analysis: the model analyzes the text and traverses the decision tree based on word presence or absence, ultimately reaching a leaf node that represents the predicted sentiment (positive, negative, or neutral)

- Fundamental machine learning algorithm
- How: build a tree-like structure where internal **nodes** represent features and **branches** represent decision rules based on those features
- During training, the model iteratively splits the data based on the feature that best separates the data points belonging to different classes
- The process continues until a stopping criterion is met $\rightarrow$ tree structure where leaf nodes represent the predicted sentiment class
- In Sentiment Analysis: the model analyzes the text and traverses the decision tree based on word presence or absence, ultimately reaching a leaf node that represents the predicted sentiment (positive, negative, or neutral)

- Fundamental machine learning algorithm
- How: build a tree-like structure where internal **nodes** represent features and **branches** represent decision rules based on those features
- During training, the model iteratively splits the data based on the feature that best separates the data points belonging to different classes
- The process continues until a stopping criterion is met $\rightarrow$ tree structure where leaf nodes represent the predicted sentiment class
- In Sentiment Analysis: the model analyzes the text and traverses the decision tree based on word presence or absence, ultimately reaching a leaf node that represents the predicted sentiment (positive, negative, or neutral)

## Decision Tree



# Advantages

- **Interpretability**: the decision-making process is easily understood by following the tree structure and rules at each node

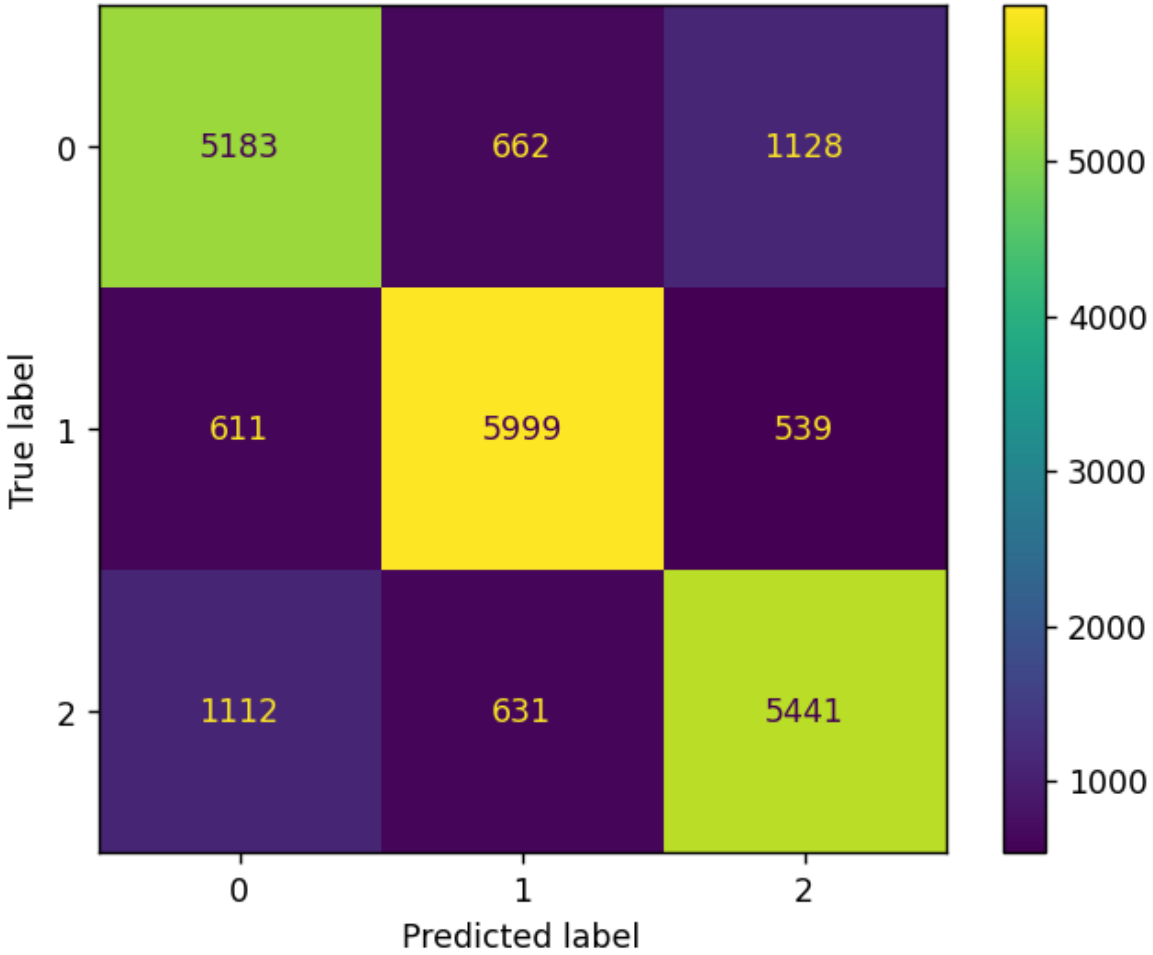
# Advantages

- **Interpretability**: the decision-making process is easily understood by following the tree structure and rules at each node
- **Handles missing data**: can effectively handle missing data points by incorporating them into the decision-making process during tree construction

# Advantages

- **Interpretability**: the decision-making process is easily understood by following the tree structure and rules at each node
- **Handles missing data**: can effectively handle missing data points by incorporating them into the decision-making process during tree construction
- **Fast training and prediction**: they are known for their computational efficiency; this can be advantageous for real-time sentiment analysis applications

# Performance



# Conclusion

| Model used | Accuracy |
|---|---|
| Logistic Regression | 86.83% |
| SVM | 77.65% |
| KNN | 55.24% |
| Naive Bayes | 68.59% |
| Decision Tree | 78.02% |